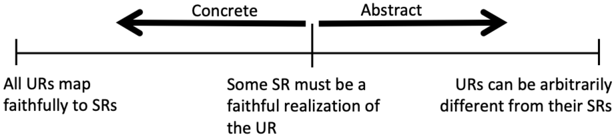
Revisiting Abstract Underlying Representations

Evidence from a learning model of probabilistic URs

Max Nelson

May 23, 2019

1. UR's do not always map faithfully to SR's. What is the extent to which UR's can differ from their corresponding SR's?



2. Kenstowicz and Kisseberth (1977) step through the continuum, conclude that the UR-SR relationship cannot be restricted

1. The less restrictions placed on the UR, the larger search space the learner is expected to navigate
2. Phonological learning and UR learning are parallel, ambiguity grows with UR space

[dɔgz]	[kæts]
/dɔg+z/→[dɔgz]	/kæt+z/→[kæts]
/dɔɣ+z/→[dɔgz]	/kæt+z/→[kæts]
/dɔg+s/→[dɔgz]	/kæt+s/→[kæts]
/dɔɣ+s/→[dɔgz]	/kæs+s/→[kæts]
...	...

3. Given a single observed surface form, what are the limits on the space of possible URs?

1. These problems may be solvable (Tesar, 2017), but is it necessary?
2. Conceptualization of URs has changed since 1977:
    - Priority Constraints (Bonet, Lloret, and Mascaró 2003; Mascaró 2007)
    - Lexical/Underlying Representation Constraints (Apoussidou, 2007; Pater et al, 2012; Smith, 2015)
    - Gradient representations (Smolensky and Goldrick, 2016; Zimmerman, 2018)
3. Assumptions that have been thrown out:
    - All morphologically related forms are derived from a single UR
    - A UR is a string of discrete phonological segments or features

- Goal: Determine to what extent, if any, previous arguments in favor of abstractness still hold
- Method:
  1. Implement a model that learns probabilistic URs in parallel with the phonological grammar
  2. Test the extent to which this model is able to learn an alternation that has previously been cited as an argument for abstractness when the model is unable to learn abstract URs
- Focus on *concrete* vs. *composite URs*
  - Concrete URs surface faithfully at least once
  - Composite URs contain a novel combination of segments or features that appear in SRs

Roadmap

1. Background
    - UR Constraints
    - Maximum Entropy grammar and optimization
    - Structural ambiguity
2. Model
    - Maximum Entropy grammar with URCs, URs as hidden structure
    - UR Constraint induction
    - Candidate generation
3. Test case - Palauan
4. Conclusions

1. Specify the UR for an input, which has no phonological content (Apoussidou, 2007; Pater et al, 2012; Smith, 2015)
2. Candidates are (Input, UR, SR) triplets
3. URs are selected in parallel with phonological optimization, allowing phonological "consequences" of a UR to affect its likelihood
   - Choosing a non-default UR and mapping faithfully is a viable repair strategy

IND + ANT	DEP	MAX	HIATUS	IND=/ə/	IND=/ən/
a. ə+ænt → əænt			∗W	L	∗W
☞ b. ən+ænt → ənænt				∗	
c. ə+ænt → ənænt	∗W			L	∗W
d. ən+ænt → əænt		∗W	∗W	∗	

1. MaxEnt (Goldwater and Johnson, 2003) is a weighted probabilistic variant of OT, weights are non-negative and violations are negative
2. Probability of a an output x given the input y:

$$p(x \mid y) = \frac{e^{\mathscr{H}_{(x,y)}}}{\sum_{\gamma \in \mathcal{Y}(x)} e^{\mathscr{H}_{(x,\gamma)}}}$$

where:

$$\mathscr{H}_{(x,y)} = \sum_{i}^{m} w_i c_i(x, y)$$

1. Learning is the process of discovering the set of weights, **w**, that produce the set of observed surface forms (in their observed proportions)
2. Optimize weights with respect to regularized negative log-likelihood:

$$\mathcal{L} = -\Big(\sum_{(x,y) \in T} \log p(x \mid y) \Big) + R(\boldsymbol{w})$$

3. Stochastic Gradient Descent:

$$\delta_{w_i} \propto \overbrace{c_i(x, y)}^{\textit{Observed}} - \overbrace{\sum_{\gamma \in \mathcal{Y}(x)} c_i(x, \gamma) p(\gamma \mid x)}^{\textit{Expected}}$$

1. Latent structure introduces ambiguity:
   - Prosodic structure: is [kapáta] an instance of [(kapá)ta] or [ka(páta)]?
   - URs: is [dɔgz] an instance of /dog+z/→[dɔgz] or /dog+s/→ [dɔgz]
2. Learner can't calculate *Observed*:

$$\delta_{w_i} \propto \overbrace{c_i(x,y)}^{\textit{Observed}} - \overbrace{\sum_{\gamma \in \mathcal{Y}(x)} c_i(x,\gamma)p(\gamma \mid x)}^{\textit{Expected}}$$

3. Expectation Maximization (Dempster et al 1977) - Use the current grammar to (probabilistically) parse ambiguous forms (Tesar and Smolensky 1998, Jarosz 2013)
4. Probability of a surface form is the sum probability of all overt-consistent forms (Pater et al 2012)

1. Trained on (Morphsyntactic features, surface form) tuples
2. Same probability function but with (ur,sr) pairs as the mapping, conditioned on morphosyntactic features:

$$p((ur, sr) \mid M) = \frac{e^{\sum_i^m w_i c_i(M,(ur,sr))}}{\sum_{(ur',sr') \in \mathcal{Y}(M)} e^{\sum_i w_i c_i(M,(ur',sr'))}}$$

(Pater et al 2012)

3. Similarly same update function:

$$\delta_{w_i} \propto c_i(M, (ur, sr)) - \sum_{(ur',sr') \in \mathcal{Y}(M)} c_i(M, (ur', sr'))p((ur', sr') \mid M)$$

Current model

UR Constraints in a MaxEnt Grammar

1. Trained on (Morphsyntactic features, surface form) tuples
2. Same probability function but with (ur,sr) pairs as the mapping, conditioned on morphosyntactic features:

$$p((ur, sr) \mid M) = \frac{e^{\sum_i^m w_i c_i(M,(ur,sr))}}{\sum_{(ur',sr') \in \mathcal{Y}(M)} e^{\sum_i w_i c_i(M,(ur',sr'))}}$$

(Pater et al 2012)

3. Similarly same update function:

$$\delta_{w_i} \propto c_i(M, (ur, sr)) - \sum_{(ur',sr') \in \mathcal{Y}(M)} c_i(M, (ur', sr'))p((ur', sr') \mid M)$$

1. Adapt Jarosz (2013)'s Expected Interpretive Parsing to MaxEnt to assign probabilistic hidden structure
2. Use the current grammar to define probability distribution over all mappings from the input to observed surface form - distribution over URs
3. Estimate violations of (M, sr) for any constraint by average violations of all candidates weighted by their probability
4. Given $Z(sr, M)$, which returns the set of candidate (ur,sr) pairs which match the overt form of *sr*:

$$\hat{c}_i(M, sr) = \frac{\sum_{\pi \in Z(sr)} c_i(M, \pi) p(\pi \mid M)}{\sum_{\pi \in Z(sr)} p(\pi \mid M)}$$

Current model

UR Constraint induction

1. URCs are the lexicon - must be learned
2. Given observed string S and corresponding meanings $M_1 \ldots M_n$
3. For every exhaustive segmentation of S that yields n nonempty substrings $s_1 \ldots s_n$:
   - For c in the set of UR constraints of the form $M_{1\ldots n}$=/$s_{1\ldots n}$/:
     - If c not in $\textsc{Con}$, add c to $\textsc{Con}$ with weight w

Example, {M1,M2}→[abc]:

Segmentation	Constraints added
a.bc	M1=/a/, M2=/a/, M1=/bc/, M2=/bc/
ab.c	M1=/ab/, M2=/ab/, M1=/c/, M2=/c/

4. Composite URCs optionally induced as new concrete URCs are added
   - Align new constraint to all existing and flip the segments at all combinations of indices of any substitutions

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in $\textsc{Con}$ for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in $\textsc{Con}$ for M_n
2. For an input $M_1 \dots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \dots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in Con for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

$\{M1_1, M2_2\}$	$\{M1\}$=a	$\{M1\}$=ab	$\{M2\}$=bc	$\{M2\}$=c
a. /**a**$_1$.**bc**$_2$/		-1		-1

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 ... M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times ... \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 ... M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times ... \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1
d. /**ab$_1$.c$_2$**/	-1		-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in $\textsc{Con}$ for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

$\{M1_1,M2_2\}$	$\{$M1$\}$=a	$\{$M1$\}$=ab	$\{$M2$\}$=bc	$\{$M2$\}$=c
a. /**a**$_1$**.bc**$_2$/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1
d. /**ab**$_1$**.c**$_2$/	-1		-1	

Current model
Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 ... M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times ... \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /**a$_1$.bc$_2$**/		-1		-1
b. /a$_1$.c$_2$/		-1	-1	
c. /ab$_1$.bc$_2$/	-1			-1
d. /**ab$_1$.c$_2$**/	-1		-1	

This is the complete space of possible URs for {M1,M2}

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c
a. /a$_1$.bc$_2$/→[abc]		-1		-1
b. /a$_1$.c$_2$/→[ac]		-1	-1	
c. /ab$_1$.bc$_2$/→[abbc]	-1			-1
d. /ab$_1$.c$_2$/→[abc]	-1		-1	

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in CON for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

{M1$_1$,M2$_2$}	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c	MAX(A)
a. /a$_1$.bc$_2$/→[abc]		-1		-1	
b. /a$_1$.c$_2$/→[ac]		-1	-1		
c. /ab$_1$.bc$_2$/→[abbc]	-1			-1	
d. /ab$_1$.c$_2$/→[abc]	-1		-1		

Current model

Candidate Generation

1. UR_n is the set of all URs specified by URCs in Con for M_n
2. For an input $M_1 \ldots M_n$:
   i. All underlying forms are generated by $UR_1 \times UR_2 \times \ldots \times UR_n$

$\{M1_1, M2_2\}$	{M1}=a	{M1}=ab	{M2}=bc	{M2}=c	Max(a)
a. $/a_1.bc_2/ \rightarrow$[abc]		-1		-1	
b. $/a_1.c_2/ \rightarrow$[ac]		-1	-1		
c. $/ab_1.bc_2/ \rightarrow$[abbc]	-1			-1	
d. $/ab_1.c_2/ \rightarrow$[abc]	-1		-1		
e. $/a_1.bc_2/ \rightarrow$[bc]		-1		-1	-1
f. $/a_1.c_2/ \rightarrow$[c]		-1	-1		-1
g. $/ab_1.bc_2/ \rightarrow$[bbc]	-1			-1	-1
h. $/ab_1.c_2/ \rightarrow$[bc]	-1		-1		-1

Simulations

1. Test the model on alternations that have been argued to require different degrees of abstract URs
2. For today, Palauan
   - *Composite* URs which contain novel combination of segments that occur across different surface forms
   - Train a model with and without the ability to induce composite URs
3. Are the composite URs necessary? Can the model learn and generalize the alternation without them?

1. Palauan alternation from Flora (1974), banner example for composite URs (McCarthy, 2011)

	-Ø	-K	-MAM	Gloss
HAB	ʔáb	ʔəbúk	ʔəbəmám	*'ashes'*
MAD	mád	mədák	mədəmám	*'eyes'*
HUR	ʔúr	ʔəríк	ʔərəmám	*'laughter'*

2. No final vowels, stress must be final, unstressed vowels must be [ə]
3. Composite analysis - underlying /ʔabu/, /mada/, and /ʔuri/

1. Constraints: IDENT(VOWEL), MAX(VOWEL), REDUCE, FINALSTRESS, *FINAL-VOWEL, *ə́
2. Overall probability correct by hyperparameters (averaged over 10 runs of 2,000 iterations with learning rate = 0.2)

Reg:	L1		L2	
Init:	Random	0.0	Random	0.0
Abstract	0.78 (0.12)	0.91 (0.03)	0.91 (0.15)	0.97 (0.01)
Concrete	0.91 (0.08)	0.89 (0.10)	0.95 (0.09)	0.97 (0.02)

Simulations

Palauan

1. Model is able to succeed with and without composite URs, in both cases best performance is L2 prior with 0.0 initialization
2. Sample grammars:

Concrete		Abstract	
Constraint	Weight	Constraint	Weight
K=/k/	7.076	HUR=ʔuri	6.236
STRESS-FINAL	6.382	STRESS-FINAL	6.161
MAM=/əmam/	5.381	HAB=ʔabu	6.057
*ə́	4.924	MAD=mada	5.908
REDUCE	4.507	REDUCE	5.365
*FINAL-VOWEL	4.427	K=/k/	5.268
HAB=/ʔab/	3.699	MAM=/mam/	5.250
MAD=/mad/	3.688	*FINAL-VOWEL	4.807
HUR=/ʔur/	3.459	*ə́	3.886
HUR=/ʔəri/	2.336	MAD=/məda/	1.77
HAB=/ʔəbu/	2.158	MAD=/madə/	1.58
MAD=/məda/	2.004	HUR=/ʔəri/	1.462
MAX(V)	1.794	HAB=/ʔəbu/	1.183
...(50)	< 0.972	...(330) < 1.040	

Simulations

Palauan

1. Model is able to succeed with and without composite URs, in both cases best performance is L2 prior with 0.0 initialization
2. Sample grammars:

Concrete		Abstract	
Constraint	Weight	Constraint	Weight
K=/k/	7.076	HUR=ʔuri	6.236
STRESS-FINAL	6.382	STRESS-FINAL	6.161
MAM=/əmam/	5.381	HAB=ʔabu	6.057
*ə́	4.924	MAD=mada	5.908
REDUCE	4.507	REDUCE	5.365
*FINAL-VOWEL	4.427	K=/k/	5.268
HAB=/ʔab/	3.699	MAM=/mam/	5.250
MAD=/mad/	3.688	*FINAL-VOWEL	4.807
HUR=/ʔur/	3.459	*ə́	3.886
HUR=/ʔəri/	2.336	MAD=/məda/	1.77
HAB=/ʔəbu/	2.158	MAD=/madə/	1.58
MAD=/məda/	2.004	HUR=/ʔəri/	1.462
MAX(V)	1.794	HAB=/ʔəbu/	1.183
...(50)	$<$ 0.972	...(330) $<$ 1.040	

Simulations

Palauan

1. Model is able to succeed with and without composite URs, in both cases best performance is L2 prior with 0.0 initialization
2. Sample grammars:

Concrete		Abstract	
Constraint	Weight	Constraint	Weight
K=/k/	7.076	HUR=ʔuri	6.236
STRESS-FINAL	6.382	STRESS-FINAL	6.161
MAM=/əmam/	5.381	HAB=ʔabu	6.057
*ə́	4.924	MAD=mada	5.908
REDUCE	4.507	REDUCE	5.365
*FINAL-VOWEL	4.427	K=/k/	5.268
HAB=/ʔab/	3.699	MAM=/mam/	5.250
MAD=/mad/	3.688	*FINAL-VOWEL	4.807
HUR=/ʔur/	3.459	*ə́	3.886
HUR=/ʔəri/	2.336	MAD=/məda/	1.77
HAB=/ʔəbu/	2.158	MAD=/madə/	1.58
MAD=/məda/	2.004	HUR=/ʔəri/	1.462
MAX(V)	1.794	HAB=/ʔəbu/	1.183
...(50)	< 0.972	...(330) < 1.040	

Simulations

Palauan

1. Abstract solution is as expected, URs are /ʔabu/, /mada/, and /ʔuri/, final vowels delete, stress is final, nonstressed vowels reduce
2. Concrete solution has two URs for each word
   - /ʔab/ (CVC) and /ʔəbu/ (CəCV)
   - Use CVC when unsuffixed or suffixed with /əmam/, reduce if necessary
   - Use CəCV when suffixed with -k
3. Is there an empirical reason to prefer the abstract solution?

1. Do the two analyses make different generalizations?
2. Phonotactic generalization to novel /kaga/ and /kagapak/, no UR constraints

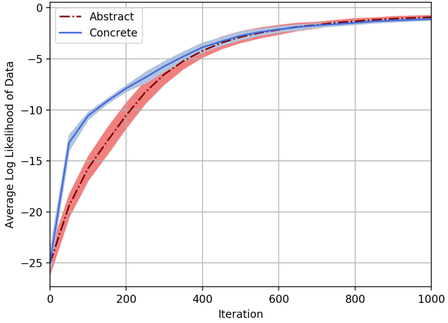
Input	Output	Concrete	Abstract
/kaga/	[kág]	0.943	0.993
	[kəgá]	0.056	0.007
	[kagá]	0.007	$\sim$0
/kagapak/	[kəgəpák]	0.981	0.977
	[kəgapák]	0.013	0.005
	[kəgápək]	$\sim$0	$\sim$ 0
	[kəgəpə́k]	0.005	0.017

3. Concrete analysis generalizes final vowel deletion despite never 'seeing' vowel deletion in the language
   - *FINAL-VOWEL is high because it motivates UR selection in the unsuffixed form, MAX(V) is pushed down by the prior

Simulations

Palauan - Discussion

1. The Palauan alternation is learnable without abstract URs
2. Including or excluding abstract URs does not change the behavior predicted by the grammar
3. Heuristics used to reduce search space - still sufficiently different (63 vs 343 constraints) to affect learning



Conclusions

1. Abstract URs are not necessary to learn and generalize the Palauan alternation
    - Including abstract URs slows learning by expanding the search space
2. In progress work not presented suggests that more abstract URs (underspecified, neutralized) are also learnable with probabilistic concrete URs
3. Previous arguments in favor of abstract URs may no longer hold

Thank you!